Documentation for Control IQ

Relating to: **CIQProc.EXE**
Date: **as of August 27, 2023**
Program Revision: **5.0.1.93**
Subject **Logic Point Verbs**

CIQProc has a 'Logic' Point Type that allows the user to perform logical and mathematical operations on the data with the CIQ program. Each logic point can utilize up to 20 variables. A variable can be any constant (On, Off, True, False), a constant value (1.23, 3.14 -2222) or a COP (Class.Object.Property) as defined in other documents.

Note that starting with CIQPric v4.0.0 The logic points can now be enabled / disabled (interlocked / not interlocked). If the point is not "enabled/interlocked" the value of the result will be what is entered into the "DisabledPosition" field. .

Note that starting with CIQProc v2.9.1.22 the Class and Properties fields are optional. The "Value" property is assumed and the process will find the correct class on its own.

AI.OAT.Value = OAT.Value = OAT

If you do not want to use the "Value" property, you must specify the desired property.

AI.OAT.Reason = OAT.Reason

Equations are not 'case sensitive'. Upper and lower case letter may be used to make the equation the most readable. A typical equation may be:

Avr(V1,V,V3,V4)

The 'equation' that is built is configured using the following parameters.

Operators

+, -, *, /	- math operators
^	- exponential
%	- percent
&	- textual concatenate
(,)	- parenthesis (nesting)

Logics

AND, OR, NOT

Conditions

<	Less than
<=	Less than or Equal to

=	Equal to
>=	Greater than or Equal to
>	Greater than
<>	Not Equal

Variables

V1 – V20 Variables are used within the equation as a "Vx" naming construct

Functions

Functions parameters are surrounded with parenthesis. For example the correct use of MIN is MIN(V1,V2,V3,12)

Functions may also be nested. MIN(V1,V2, AVG(V3,V4,V5)) is acceptable.

Numerical

AVR	Returns the average of the series	AVR(V1,V2,V3…)
ABS	Return the absolute value	ABS(-123) [123]
CtoF	Converts Centigrade to Fahrenheit	CtoF(V1)
FtoC	Converts Fahrenheit to Centigrade	FtoC(V1)
EXPORT	Export Data to an Excel File	
IMPORT	Import Data from an Excen File	
INT	Rounds down the result to the nearest integer value	
LIMIT	Limits the result(V1,V2,V3) V1 is the value,V2=min V3=max	
MIN	Returns the minimum value in the series Min(V1,V2,V3)	
MAX	Returns the maximum value in the series Max(V4,V,V3)	
ROUNDUP	Rounds up the result to the nearest integer value	
Scale	Reset Equation, there are 6 possible	

Scale (FeedBack, Y1, X1, Y2, X2 [,clamp])

Typically this would be something like
Scale(V1, 40, 0, 90, 100)
Where
V1=AI.OAT.Value

The output is scaled between 0 and 100 as the outside air ramps between 40 and 90. This is spoken as "When OAT = 40 make me 0, when OAT = 90 make me 100)

The clamping is default between X1 and X2, however with an optional 6 value you can tell it **not** to clamp

Scale(V1, 40, 0, 90, 100**, 0**) [1 is implied]

SELECT	Like a lookup table, it selects the Nth variable based on the first variable [0 based]. SELECT(V1,10,20,30,40,50). If V1=0 then the result is 10. If V1=3 then 40 is returned.
SQL	Execute an SQL statement directly (see details below)
SQR	Returns the square root of the SQR(V19 *23)
SubMn	Minimum Value of all point in a Subsystem / PointType
SubMx	Maximum Value of all points in a Subsystem / PointType
SubAv	Average Value of all points in a Subsystem / PointType For SubMn, SubMx and SubAV there are 5 values that must be set V2 – Minimum Reasonable Value “45” V3 – Maximum Reasonable Value “95” V4 – Value to Return when no match “72” Usage = SubMn(P1Asic1s,ASIC1,V2,V3,V4) or each variable can defined as a “V” variable
Switch	Set a bit on or off. There are 4 variables that must be used V1 – The controlled point that this will effect V2 – The feedback to determine if it is true V3 – The Value at which Switch will go true V4 – The Value at which Switch will go false Usage=Switch(DO.Heater.Value, AI.WallSensor.Value. 70, 72)
SPvsAct	Compare Setpoint vs FeedBack V1 – Setpoint V2 – Feedback V3 – Deadband V4 – Deadband (optional) greater than V3 V5 – Deadband (optional) greater than V4 If V4 and V5 are not defined – returns -1,0,1 If V5 is not defined – returns -2, -1, 0, 1, 2 If V4 and V4 are defined – returns -3, -2, -1, 0, 1, 2, 3

<u>Date / Time</u>

DAY	Returns the day of the month (1-31)
DOW	Returns the day of the week (1-7) (1=Sunday)
MONTH	Returns the month number (1-12)
YEAR	Returns the year number 2004
NOW	Returns the current date time as a decimal
HOURS	Returns the current hour (0-23)
MINUTES	Returns the current minute (0-59)
SECONDS	Returns the current second (0-59)

Logarithmic

Most logarithmic functions are supported by CIQProc, but are not discussed here because of their low probability of use. The supported functions are: ARCSINH, ARCCOSH, ARCTANH, ARCSECH, ARCCSCH, ARCCOTH, ARCSIN, ARCCOS, ARCSEC, ARCCSC, ARCCOT, ATN, CLG, CSC, CSCH, COS, COT, COTH, LOG, SEC, SECH, SIN, SINH, TAN, TANH

SQL – Execute an SQL statement directly

This function allows the users to define specific queries against the database that will resolve in a single value. Structured Query Language (SQL) is very robust, yet very complex, so this document will not address all of the results that you can achieve. SQL is also very unforgiving for syntax errors so it should not be done without some prior knowledge or testing (like with Navicat).

A simple SQL statement to find the average Outside Air Temperature would look like:

SELECT AVG(Value) FROM points_values WHERE alias like'%oat'

Where

Points_values is the table to collect the information from
Value is the field (column) to get the data from
AVG() is a mySQL average function
Like '%oat'
% means an leading characters (wild card)
Oat means it must end in 'oat'
(notice the single quotes after the LIKE clause

So this will average AH11OAT, AH12OAT, CH1OAT
But it will not include AH15OATemp

You could use "Like '%oat%' " and that would get the AH15OATemp, but be careful because it would also get BOATRAMP if that were an available alias.

To use this in a LOGIC point, surround the statement with the function 'SQL(), so the above would be entered as an equation like:

SQL(Select AVG(Value) from points_values where alias like '%oat')

Some simple rules for use:

- No part of the SQL statement is case sensitive
- Then minimum calculation timer is set at 60 seconds for SQL functions

- You cannot have 'other' functions in the same equation like:

  SQL(Select AVG(Value) from points_values where alias like '%oat') *** 123**

- A '%' in the like clause is a wild card of 0 to any number of characters
  %oat will work for OAT, AH11OAT, Chiller1OAT
  %oat% will work for OAT, AH11OATemp, Boat

- A '_' (under score) is a wild card for only 1 character
  VAV6_1__ will work for VAV62101, VAV63142 etc (note two underscores at the end). It will not return VAV62101ST
- If the Logic point cannot evaluate the SQL, it will return a -999 value

A few examples:

SQL(Select Avg(Value) from points_values where Alias like '%oat')
Will bring back the average of all points that match that alias
Remember it is ALL points, even those not communicating

SQL(Select Avg(Value) from points_values where Alias like '%oat' and Reason <> 31 and Reason <> 33)
Reason 31 means the point did not read on the last scan
Reason 33 means it hasn't read for 30 minutes (or more)

SQL(Select Min(Value) from points_values where Alias like 'VAV6_1__' and Reason > 100)
All valid reads on ASIC/1s will result in a Reason > 100
This will return the minimum zone temperature

SQL(Select Count(State) from points_values where Alias like 'VAV6_1__' and State=1 or State = 4)
This will return all VAVs in Phase 6, Fan 1 (61) where they are occupied or in Afterhours (motion detected as well)

State=0	unoccupied
State=1	occupied
State=2	night setback
State=3	morning ready (morning warmup)
State=4	afterhours

There are many functions that can be used after the Select statement. A description and syntax of these can be found at:

http://dev.mysql.com/doc/refman/5.1/en/functions.html

The above examples show how to use SQL within the Control IQ database. It is also possible to retrieve data from foreign database. The foreign database must be defined in the ODBC drivers. If it is a simple DSN, you can put the DSN name in V1 of the logic point. If it is more complex (or longer than 30 characters) you can add the ODBC parameter to the Equations as follows:

SQL(Select Count(OrganizationName) from Organizations),
ODBC(DSN=IssueTrak-CIQ;uid=sa;pwd=thePass)

Excel Files

Excel is now supported. You can now read/write to excel files. This requires Microsoft Excel to be installed on the machine that CIQProc is running on.

It helps to know some Excel syntax.

The syntax is not case sensitive but it is VERY specific.
: (colon) is the delimiter for a range of cells
! (exclamation) is the delimiter for a spreadsheet number or name
Variables (V1-V20) must be enclosed in parenthesis

Assuming today is October 23, 2013 at 17:48 PM
A 'read' aka IMPORT can only be a single cell

To read you need to setup the Equation.

IMPORT(FileName,Cell)	**- basic format**
IMPORT(c:\myexcel.xls,B4)	**- read the contents of B4**
IMPORT(c:\myexcel.xls,B(V1))	**- read the contents of Colum B at Row defined by V1**
IMPORT(c:\my?month.xls,B4)	**- read the contents of B4 from "My_October.xls"**
IMPORT(c:\myexcel.xls,Sheet2!B4)	**- read the contents from Sheet2 B4**
IMPORT(c:\myexcel.xls,3!B4)	**- read the contents from 3rd sheet B4**
IMPORT(c:\myexcel.xls,Meter!B7)	**- read the contents from Sheet called "Meter"**
IMPORT(c:\myexcel.xls,(V1)!B4)	**- read from a sheet define in V1**
IMPORT(c:\?Year.xls,(V1)!B(V2))	**- read from a file called "_2013.xls" - from a sheet define in V1 – txtMonth - from column B at row V2 - DayOfMonth**

But a 'write' aka EXPORT can write to a range of cells

EXPORT(FileName,Cell,VALUES()) **- basic format – values from V1-V20**
EXPORT(c:\test.xlsx,B3,Values(V1)) **-write to a single cell**
EXPORT(c:\test.xlsx,B3:B6,Values(V1,V2,V3,V4))
EXPORT(c:\test?Month.xls,B3:B6,Values(V1,V2,V3,V4))
EXPORT(c:\test?Month,2!B3:B6,Values(V1,V2,V3,V4))
EXPORT(c:\test?Month,Sheet2!B3:B6,Values(V1,V2,V3,V4))
EXPORT(c:\test?Month,Sheet2!B3:C4,Values(V1,V2,V3,V4))
EXPORT(c:\test?Month,Sheet2!B(V1):C(V2),Values(V3,V4,V5,V6,V7,V8,V9,V10))

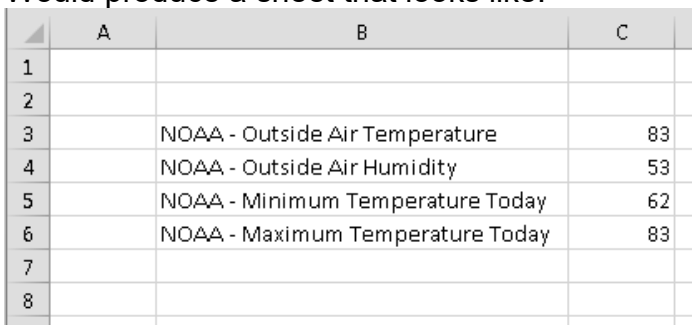
Cells are written in columns then rows. In the above example

B3=V3
C3=V4
B4=V5
C4=V6
B5=V7 etc

Consider these values for:

V1=3 (B3)
V2=6 (C6)
V3=NOAAOAT.Name
V4=NOAAOAT.Value
V5=NOAAOAH.Name
V6=NOAAOAH.Value
V7=NOAAMINT.Name
V8=NOAAMINT.Value
V9=NOAAMAXT.Name
V10=NOAAMAXT.Value

Would produce a sheet that looks like:



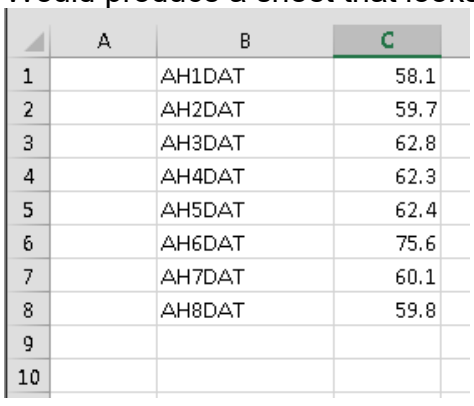
	A	B	C
1			
2			
3		NOAA - Outside Air Temperature	83
4		NOAA - Outside Air Humidity	53
5		NOAA - Minimum Temperature Today	62
6		NOAA - Maximum Temperature Today	83
7			
8			

But wait, there's more! Logic Points can also write to an Excel sheet based on the result of an SQL query.

EXPORT(FileName,Cell,SQL())

EXPORT(C:\DATemps.xls,B1:C10,SQL(Select Alias,Value From Points_Values where Alias='%DAT'))

Would produce a sheet that looks like:



	A	B	C
1		AH1DAT	58.1
2		AH2DAT	59.7
3		AH3DAT	62.8
4		AH4DAT	62.3
5		AH5DAT	62.4
6		AH6DAT	75.6
7		AH7DAT	60.1
8		AH8DAT	59.8
9			
10			

File Variables: note the underscores.

```
?Today      - _2013_10_23
?YYMMDD     - _131023
?YYMM       - _1310
?Year       - _2013
?Month      - _October
?Day        - _23
?Hour       - _17
```

Logic points will create the file if it does not exist. This allows for dynamic file creation based on date/time as ControlIQ continues to run

The minimum calculation time for a logic point to import or export to an excel file is 1 minute.

Note CIQ will read the excel file if it is opened by someone else, but it will not write to the file until it has exclusive write access. (Don't leave your files open)

User Variables

There are times that users want to enter data into Control IQ for trending, alarming and reporting, where that data is not read directly by Control IQ. There are a number of methods to do this.

- Use the USERxx variables in the preferences table
- Read the data from am Excel file (requires Microsoft Excel installed)
- Use a Logic point.

To use a Logic Point, our recommendation is the following:

- Subsystem Set to 'None'
- Include In Setpoint Set to 'True'
- Equation Set to 'V1'
- V1 Set to 'my.setpoint'
- Optionally, setup the points that you want to manually enter data into their own group.
- In Tabular Data, right click and select "Change Setpoint"
- In View, the data can be change by setting Left-Click to spadjust?alias={alias}